

- definition of a software architecture implementing the formal architecture,
- definition of implementation mechanisms of the formal specification, where the rational agent is thus capable of conversing with another agent or with a system user through any communications media (vocal or written: computer screen, keyboard, mouse, etc.).

The different components of the formal model are unified in the same formal framework (logical theory) with the same formalism. The generic nature of the mechanisms and principles gives the model a degree of independence with regard to the application, the communications media and the language.

- The definition of implementation mechanisms is realized so as to obtain a direct correspondence between these mechanisms and the aforesaid model.

The formal specification of the different components of the formal architecture and their combination implies a level of rationality axioms, a level of communication axioms, and a level of cooperation axioms.

- The definition of the software architecture implementing the formal architecture includes: a rational unit including an implementation layer of the rationality axioms level, an implementation layer of the communication axioms level, an implementation layer of the cooperation axioms level, corresponding respectively to axioms of the formal model.

- The software definition implementing the formal architecture includes besides:

- a generation module and comprehension module implementing a natural language level layer.

The rational unit, the generation module, and the comprehension module implement the implementation
5 mechanism of the formal model.

The generation module is suitable to transcribe a logical statement produced by the rational unit in natural language for system usage.

The comprehension module is suitable for
10 interpreting the user's statement into a logical statement comprehensible to the rational unit.

The invention also has as its object a conversational rational agent placed as a kernel of a dialogue system, and/or as element (agent) of a
15 multiagent system, including:

- a definition of a conceptual architecture,
- a formal specification of the different components of this architecture and their combination permitting a formal model to be
20 obtained,

mainly characterized in that it includes:

- a definition of a software architecture implementing the formal architecture,
- a definition of implementation mechanisms of the
25 formal specification realized by a rational unit that comprises:
 - data including predefined axiom schemes and application-dependent axiom schemes,
 - an application-dependent knowledge base
30 including a semantic network and interconcept distances,

- an inference engine to implement formal specification mechanisms by means of data and the knowledge base so as to be able to receive a logical statement, understand it, and be able to provide a logical statement in response,

According to another characteristic, the data include implementation data of a formal model including:

- an implementation layer of rationality axioms, an implementation layer of communication axioms, and an implementation layer of cooperation axioms, corresponding respectively to the axioms of the formal model.

According to another characteristic the agent includes besides:

- a statement generation module in natural language using a logical statement coming from the rational unit and a comprehension module to provide a logical language statement to the rational unit from a natural language statement; these modules thus implement a communication level layer in natural language.

The invention also has as its object an information server, including the means to implement a man-machine dialogue system the kernel of which rests on the implementation of a conversational rational agent as previously defined.

The invention also concerns a multiagent system including the communicating agents, each agent

including the means to implement an interaction, the system including at least one agent the kernel of which rests on an implementation of a conversational rational agent as described previously.

5 Other specifics and advantages of the invention will appear clearly to the reader from the description made below by way of non-restrictive examples and regarding the figures of which:

- 10 ▪ figure 1 represents the software architecture of a conversational rational agent,
- figure 2 represents the architecture of the rational unit and its knowledge base,
- 15 ▪ figure 3 represents in a more detailed manner the software architecture of a conversational agent as a kernel of a dialogue system (specifically oral),
- figure 4 represents an architecture showing a conversational rational agent as the kernel
- 20 of a multiagent system.

The conversational rational agent approach that has been executed by the filer and that has been the object of publications, has been guided by principles

25 of rationality, communication and cooperation, formalized in the theory of the rational interaction. For this purpose you may refer to the previously mentioned publications that pertain to the "conversational rational agent" approach.

30 The definition of the conceptual architecture of a conversational rational agent is given in the appendix

of the description. This definition was the object of a publication in "Conseil Scientifique de France Télécom" (French Telecom Scientific Council), Technical Memo n°8: Smart Interfaces and Images" October 1996, pp. 37-61.

For the following you may refer to the diagram of figure 1.

According to the invention, the filer has implemented these principles by means of a rational unit 100 that constitutes the kernel of each agent which determines its reactions to external events, whether these are prompts (requests, answers, confirmations etc) from human users or prompts from other software agents (which is the case when an agent is the kernel of a multiagent system).

The rational unit 100 is driven by a inference engine that automates reasoning according to the rational interaction principles that the agent's programmer can adapt or enrich, in a declarative manner, according to the task to accomplish.

For this purpose, as will be specified below, these reasoning processes are guided by predetermined axiom schemes (enumerated in appendices) input to the unit by the agent's programmer in a declarative manner as a function of the tasks that said agent must fulfill.

Figure 1 illustrates an agent's software architecture diagram in the case where such an architecture is applied to the constitution of a dialogue system with the users.

Figure 1 therefore represents architecture of an

agent in interaction with a user, through as will be seen, a comprehension module 150 and a generation module 160. This architecture corresponds to a first possible application family that is the (convivial) interaction user service.

In order to permit dialogue with the users, the rational unit 100 is connected to an interface to the outside 140.

This interface thus includes the comprehension module 150 that receives statements in natural language and interprets these statements in a logical statement that acts as input to the rational unit 100.

The interface also includes the generation module 160 that expresses the reaction of the rational unit 100 in a natural language statement sent to the user.

In this framework, the rational unit 100 is the central entity of the service to render whatever information is required (train timetables, stock exchange trends, weather reports...) reservations or purchases or the look-up of information on the Internet.

The cooperation principles installed in the rational unit and natural language processing modules assure a convivial interaction with the user. This interaction can take place directly by speech by incorporating the dialogue system thus formed by the speech recognition and synthesis modules (not represented in this figure).

However, the rational unit 100 can by itself constitute the kernel of a self-contained software agent. In this framework, this unit interacts with

09869205-091701

other software agents by means of an inter-agent communications language such as the "Agent Communication Language" (A.C.L. adopted as a standard by the FIPA consortium).

5 The services that the agent can render are then for example transactions on the electronic markets, network administration tasks, broadcasting information.

 These two forms of interaction can be combined in such a way that after the interaction in natural
10 language with an user, an agent fulfills any task whatsoever by interactions in ACL language with other software agents distributed on the public or private networks.

 We are now going to detail the software
15 architecture functionalities of the rational unit 100, this detailed architecture been illustrated by the diagram figure 2.

 First of all, the rational unit 100 implements the principles coming from rational interaction theory the
20 objective of which is to formalize and automate the rational behavior of an agent in interaction situations with other agents or service users.

 This theory depends on two major notions the notion of modal logic on the one hand, where the
25 objective is to allow the mental attitudes of self-contained agents to be represented and, the notion of language acts on the other hand, where the objective is to specify the effects of communication on the mental attitudes of agents.

30 The contribution of the theory of rational interaction is to formalize these two domains and in

particular their interaction.

An agent's state at any given time in a communicating exchange is thus characterized by a set of mental attitudes.

- 5 The mental attitudes that may be represented are for example the *belief* usually noted by the operator K and the intention noted by the operator I.

These operators are indexed by the agent whose mental attitude they represent.

- 10 In a dialogue with the system s and the user u, Ks designates the belief operator for the system and Ku the same operator for the user.

- Language acts that may be modeled are among others, the information and request acts. The modeling
15 consists of a logical statement or logical language, for example:

Ks Iu Done (<s, Inform Si(u,p)>)

This logical statement is translated as follows:

- the system s knows (operator K) that the user u
20 has the intention (operator I) that a certain communicative act is to be performed, that is s informs u whether a certain proposition p is true or false,

or in shorter form: "s knows that u wishes s to report to him/her on the truthfulness of p."

- 25 The logical language thus defined permits expressing general principles of behavior that will determine the reactions of the rational unit.

An agent s will be cooperative if it adopts the intentions of the user u. This can be expressed thus:

- 30 Ks Iu $\phi \rightarrow$ Is ϕ

09869205-091701

Such axiom schemes on a very general scale are already predefined by the theory of the interaction and form part of an agent's rational unit.

However, the programmer of the rational unit can
5 define new, more specialized, schemes for a given application.

The set of schemes guide the reasoning of rational unit 100 and therefore its reactions to prompts from the environment.

10 The calculation of these reactions is made by the inference engine 101.

The rational unit 100 therefore include a data set 102 that includes the axiom of the formal model of the conversational rational agent. This data implements the
15 rationality layers of the agent's communication and cooperation.

Environmental prompts, for example users requests, or those from other software agents are transmitted to the rational unit 100 in the form of a ACL logical
20 statement from the theory of rational interaction.

The inference engine 101 is suitable for calculating the consequences of this statement and in particular possible responses or precise requests to be provided to the interlocutor (which may act as a
25 software agent or a human user) but also for other non-communicative actions.

Specifically for a given statement, the inference engine 101 examines whether or not it has a behavior principle available that it can apply to this statement
30 to deduce its logical consequence or consequences. This procedure is then applied to these new consequences

09869205-091701

until the possibilities are exhausted.

From among all these consequences, the inference engine 101 isolates communication or other actions that it must perform and that then form the reaction of the
5 rational agent.

The first stage of the inference procedure is placing the processed statements in normal form so as to assure that each statement is only introduced in one given syntactic form so as to be able to assure the
10 sorting and the comparison of statements.

This placement in normal form permits equally ensuring a first application of the simple principles of reasoning.

The inference procedure for each statement
15 processed then consists of verifying whether this statement corresponds to one of the axiom schemes 102 that code for the rational behavior principles retained.

This verification mechanism rests mainly on the
20 unification operation of the Prolog language.

The set of these axiom schemes can be modified by the programmer of the rational unit 101 who may delete or add axiom schemes or alternatively modify those already existing to fine tune the rational unit's
25 behavior.

These modifications can take place dynamically. In this case the rational unit modifies its behavior accordingly.

The inference procedure set is controlled so that
30 the rational unit does not enter in an infinite reasoning loop. The termination of this procedure is

thus assured.

The reasoning of the rational unit thus takes its support from a data set that depends heavily on the application sought by the rational agent.

5 When an agent is desired to provide train timetables, it must have available the data on stations and the connections between them as well as the temporal notions.

10 The data set is structured in a knowledge base 120 in the form of a semantic network.

15 The semantic network 120 permits expressing notions of classes and subclasses, and instantiation of each class. It also defines the notion of relation between classes that applies to the different instances of classes.

20 For example, for an appointment type of application, the semantic network 120 will include at least the classes "nobody" (whose instance will be the set of known people in the appointment schedule) and "function" (whose instances will be the known functions).

These two classes are in the "*the-function-of*" relation.

25 To indicate that the person Jean is in Advertising, the semantic network includes the Prolog fact: *the-function-of* (Jean, Advertising agent).

30 An access to the semantic network 120 is realized at any times during the inference procedure when the consequences of the inference depend on the kind of data.

In the appointment application, for example, if

the user asks what is Jean's profession, the response of the rational agent is going to depend on its querying the semantic network 120.

The semantic network 120 can also have notions of
5 semantic environment that are partially useful to produce cooperative responses of the rational agent.

It consists in assigning relative distances to the different processes of the semantic network, these distances are determined according to the application
10 at the time of the creation of the semantic network.

The instances of the semantic network 120 are thus projected in a metric space where its dimensions are the different network relations.

For example, the advertising function will
15 probably be determined semantically as closer to the function marketing engineer than mechanic.

This construction permits us to perform two symmetrical operations called constraint relaxation and restriction.

20 The relaxation of constraints aims to give some answers close to the initial request when the answer to the latter does not exist.

So, for example, one asks the appointment schedule who the marketing engineers are and [is answered] that
25 none exist, the inference procedure can trigger a relaxation stage so as to give the coordinates of advertising agents.

The restriction, aims on the other hand to seek how to specify a request more precisely that is too
30 general (broad). If there are 500 advertising agents recorded in the appointment schedule, a restriction

stage will give the most discriminate dimension of this too broad a set (for example the company or work of the advertising agent) in order to be able to ask a pertinent question so as to identify the user's request.

Figure 2 also permits illustrating that the rational unit 100 of a rational agent include both a generic part independent of the application and an application dependent part.

The inputs and outputs of the rational unit 100 can be considered as statements in ACL. The conversion under normal form of these statements and the inference procedure are independent of the application as well as a majority of axiom schemes that guide the system behavior. However some among them are especially adapted or created for the application as well as the semantic network that contains application data. The network 120 must, most the time, be able to respond to restriction and/or relaxation requests by the inference engine 101 as may be seen in more detail hereunder.

In this case the network must, have notions available of semantic distance between the instances as was stated.

The diagram figure 3 illustrates in more detail an agent's software architecture according to the invention.

The comprehension module of the natural language interprets a user's statement into a logical statement comprehensible by the rational unit 100.

The vocabulary processed by this module depends in part on the service the rational agent must render.

This dependent part of the application is mainly present in the semantic network 120 of the rational unit, which explains that the comprehension module 150 uses numerous data coming from the semantic network
5 120.

The comprehension module 150 is suitable to take the user's statement into account as a suite of small syntactic structures (most often words) that will each activate one or several (in the cases of synonyms)
10 notion(s) coming from the semantic network 120.

The link between the input vocabulary of the user and the semantic network 120 is done therefore by means of a concept activation table 131 that indicates what semantic notion(s) correspond to words (or suite of
15 words) of the vocabulary.

These activated notions depend in part on the desired application, but represent also much more general concepts such as negation, intention and user's knowledge, existence, cardinalities, etc.

20 The comprehension module therefore has available an activated concept list (indeed several in the case of synonyms).

It is suited to transform them into a logical statement formed by a semantic completion procedure.
25 This procedure starts from the hypothesis of semantic connectivity of the user's statement, that is to say the concepts that it has evoked are in relationship to one another.

The module 150 is suitable for connecting between
30 them, by the relationships present in the semantic network, including therein if necessary the creation of

09869205-091701

new concepts.

The procedure determines notions implied in the user's statement.

It is possible to indicate that some relationships
5 in the user's statement are incompatible with one another. The search possibilities of the completion procedure are thus controlled.

The semantic completion resorts to a weighting function 132 that permits fixing a digital weight for
10 every relationship of the semantic network, thereby representing the verisimilitude of vocations by this relationship.

In this manner, the completion procedure takes in account a notion of verisimilitude when it must
15 determine what concepts are implied by the user. In case of synonyms, these weights also permit associating a cost with each possible interpretation. Thus, only one statement will finally be retained by the comprehension module—the lowest-priced one.

20 To facilitate the semantic completion it is also possible to specify that some concept - relation or concept - concept couples are implicit. If only one of the concepts has been evoked and even though the statement studied is related, the corresponding
25 relation will be added because it is implied in an almost certain manner.

For example, in an application providing the progress of the stock market, the statement "I would like CAC 40" will be completed in an implicit manner by
30 "I would like the progress of CAC 40."

On the other hand, the comprehension module 150

must take into account the context of the user's statement.

For that, it has available the concepts previously evoked at the time by the user and by the agent itself
5 in its answers to the user. A part of these can be used therefore at the time of the completion procedure.

There again, it will be indicated for all relations of the semantic network, whether it is pertinent to keep them in context.

10 The comprehension module 150 does not use a syntactic or grammatical analyzer. This allows it to correctly interpret syntactically incorrect statements, which is particularly important in an oral dialogue context (and for voice recognition usage), since the
15 syntax of spontaneous speech is much freer,

Moreover, since the analysis is made by small syntactic components, it is not necessary to construct a grammar that is going to try to predict in advance the set of possible statements by the users.

20 Finally, the sole part dependent upon the user's language is the table joining the users vocabulary to the concepts of the semantic network.

The semantic data of the network represent in effect universal notions. This point particularly
25 facilitates the transfer of one application from one language to another language.

The generation module 160 accomplishes the inverse task of the comprehension module. It is capable of transcribing a sequence of communicative acts produced
30 by the rational unit 100 into a statement in the user's natural language.

09869205-091701

The generation procedure operates in two phases.

The first phase consists in making all decisions for the linguistic choice offered in order to verbalize the communicative act sequence provided in the input module.

For that the generator 160 uses the dialogue context among other elements to construct the statement most suited to the current situation.

Thus, in an appointment application, the module 160 should make a customized choice between equivalent formulations such as "Jean's telephone number is," "Jean's number is," or "his number is," "it is..." according to the dialogue context.

The objective of this first phase is to construct an intermediate representation of the statement by using a notion of abstract linguistic resources 133. An abstract linguistic resource represents either a lexical resource 135, for example the common names, verbs, adjectives, or a grammatical resource, that is to say the syntactic structure.

The second phase uses this abstract representation to construct the definitive statement.

It concerns a processing stage that only requires the strict application of grammar rules. Among these phenomena are found, for example, the determination of the order of the statement's constituents, the agreement between these constituent and the verbs declinations.

Comprehension 150 and generation 160 modules use as their input format respectively the output, of the written texts.

09869205.091701

If a vocal interaction is desired with a rational agent, it is necessary to attach to it the speech recognition and synthesis modules.

The recognition module 170 records the user's vocal
5 signal in a text corresponding to the pronounced statement. This module 170 is essential for example when a rational agent is used as a telephone server: then the only possible interaction is vocal.

The rational unit including the semantic network
10 that models the data that this latter manipulates, together form the kernel of a software agent.

As such, this agent can communicate through the network for example with other software agents.

The ACL communication primitive defined by the
15 rational interaction theory constitutes a communication language between agents that allow them to realize unambiguous interaction.

The agents formed by a rational unit 100 and their semantic network 120 without their interaction
20 components in a natural language (modules 140 and 150) are particularly well suited to the use of the ACL communication language between software agents to form multiagent systems as represented in figure 4.

The invention has been implemented with a SUN
25 Ultra1 workstation (provided with a 166 Megahertz processor) and a SUN Ultra2 work station (possessing two 64 bit processors and clock frequency of 300 Megahertz).

Active memory is used where the size can be around
30 32 Megabytes minimum. The maximum response time of the system is 2 seconds on the Ultra2 platform and 5

09869205.091701

seconds on the Ultral. The connection with a digital network can be made by means of a integrated digital services network interface card IDSN-Basic Rate Interface.

- 5 The three modules that have been described, the comprehension 150, the generation 160 and the rational unit 100 were implemented in Prolog (Quintus version 3.3 for Solaris 2.5). The communication between the different modules and speech recognition and synthesis
- 10 systems is performed by a program written in "C" language, a prototype of the invention has been developed under Solaris, but a version not including speech recognition and synthesis modules has been executed under WINDOWS NT 4.0.

ANNEX

ELEMENTS OF THE LOGICAL FORMALIZATION FRAMEWORK: FORMAL SPECIFICATIONS

- 5 The concepts of mental attitudes (belief, uncertainty, intention) and action that are manipulated here are formalized within the framework of a modal logic of the first order (cf. the publication: Sadek 91a, 92, for details of this logic). The aspects of the
- 10 formalism are introduced briefly, which are used in the exposition that follows. In the following, the symbols \neg , \wedge , \vee , and \Rightarrow represent the classic logical connectors of negation, conjunction, disjunction and implication, and \forall and \exists , the universal and existence quantifiers; p
- 15 represents a close formula (denoting a proposition), ϕ , ψ , and δ the formula schemes, and i and j (sometimes h) the schematic variables denoting agents. It is noted that $\models \phi$ indicates the fact that the formula ϕ is valid.
- 20 The mental attitudes considered as semantically primitive, namely belief, uncertainty and choice (or preference) are formalized respectively by the modal operator K , U and C . Formulae such as $K(i\ p)$, $U(i, p)$, and $C(i, p)$ can be read respectively as " i believes (or
- 25 thinks that) p (is true), " i is uncertain of (the truth of) p " and " i wants p to be actually true." The logical model adopted for the operator K takes into account of the interesting properties for a rational agent, such as consistencies of its beliefs or its capacity for

09869205-091701
107160-5026980

introspection formally characterized by the validity of logical schemes such as

$$K(i, \phi) \Rightarrow \neg K(i, \neg \phi),$$

$$K(i, \phi) \Rightarrow K(i, K(i, \phi)), \text{ and}$$

5 $\neg K(i, \phi) \Rightarrow K(i, \neg K(i, \phi)).$

For the uncertainty, the logical models also guarantee the validity of desirable properties such as, for example, the fact that an agent cannot be uncertain of its own mental attitudes ($\models \neg U(i, M(i, \phi))$, where M belongs to $\langle K, \neg K, C, \neg C, U, \neg U \text{ etc.} \rangle$). The logical models for the choice causes properties such as the fact that an agent "assumes" the logical consequences of its choices.

15 $(\models (C(i, \phi) \wedge K(i, \Rightarrow \psi)) \Rightarrow C(i, \psi)),$

or that an agent cannot not choose the courses of events in which it thinks it finds itself already

$$(\models K(i, \phi) \Rightarrow C(i, \phi)).$$

The attitude of intention that it is not semantically primitive, is formalized by the operator I that is defined (in a complex manner) from operators C and K . A formula such as $I(i, p)$ can be read as "i has the intention to realize p ." The definition of the intention imposes on an agent not to seek to attain that which it thinks is not already attained

25 $(\models (I(i, \phi) \Rightarrow K(i, \phi))),$ and guarantees the fact that an agent does not have the intention of carrying out effects outside of its intentions (thus "to have the intention to connect to a network and to know that it can contribute to cluttering it, does not imply

09860205-093703

(necessarily!) to have the intention of contributing to clutter the network").

In order to permit the reasoning on the action, event sequences are included, in addition to individual objects and agents. The language contains terms (in particular variables e, e_1, \dots) that run through the set of these sequences. A sequence can be formed from only one event (that can be an empty (null) event). In order to be able to speak of complex plans, the events (or sequences - $\alpha_1 : \alpha_2$, or non deterministic choices $\alpha_1 | \alpha_2$; the schematic variables, $\alpha, \alpha_1 \alpha_2 \dots$, are used to denote expressions of action. Operators are also introduced *Feasible*, *Done* and *Agent* (i, α) such as the formulae *Feasible*(αp), *Done*(αp) and *Agent*(i, α) signify respectively (the action or the expression of action) α can take place after which p will be true, α has just taken place before which p was true, and i denotes the unique agent of events appearing in α .

A fundamental property of the logic proposed is that agents modeled are in perfect agreement with themselves with regard to their own mental attitudes. Formally, the scheme $\phi \leftrightarrow K(i, \phi)$, where ϕ is governed by a modal operator formalizing a mental attitude of the agent i , is valid (Sadek 91a, 92).

The following abbreviation are used, where True is the promotional constant always true:

$$\begin{aligned} \textit{Feasible}(a) & \equiv \textit{Feasible}(a, \textit{True}) \\ \textit{Done}(a) & \equiv \textit{Done}(a, \textit{True}) \end{aligned}$$

$Possible(\phi) \equiv (\exists e)Feasible(e,\phi)$

$KIf(I,\phi) \equiv K(i,\phi) \vee K(i,-\phi)$

$Kref(i, \alpha \delta(x)) \equiv (\exists y)K(i, \alpha \delta(x)=y)$: the agent i knows the (or the objects that is one) δ , where α is the description operator defined as (producer of terms) such that:*

$\phi(\alpha \delta(x)) \equiv \exists y \phi(y) \wedge \delta(y) \wedge \forall z(\delta(z) \Rightarrow z=y)$

$Ureff(i, \alpha \delta(x)) \equiv (\exists y) U(i, \alpha \delta(x)=y)$

10 RATIONALITY PRINCIPLES AND ACTION MODEL

Two rationality principles establish the link between an agent's intentions and its plans and actions (Sadek 91a, 91b). The first principle stipulates that an agent cannot have the intention of realizing a given proposition without by itself having the intention that is formed by one of the actions that it thinks it has for the purposes of the proposition in question, and for which there is no particular objection that it be done. Formally, it is expressed by the validity of the following scheme:

$$I(I,p) \Rightarrow I(Done(a_1|...|a_n))$$

where a_k are all actions such as:

- p is the rational effect of a_k (i.e. the reason for which a_k is planned);
- the agent i recognizes the action a_k : $Kref(i,a_k)$
- $\neg C(i, \neg Possible(Done(a_k)))$

The second principle stipulates that an agent that has the intention a given action be done, necessarily adopts the intention that this action is feasible, if it does not think so already; which is expressed

formally by the validity of the following scheme:

$$I(i, Done(a_j)) \Rightarrow K(i, Feasible(a_j) \vee I(i, K(i, Feasible(a_j))))$$

5 The solution to the problem of the effect of an
action is linked directly to the expression itself of
rationality principles. We consider that if one cannot
predict the actual effects of an action, we can however
say (in a valid form) what we expect from the action,
10 otherwise stated, the reason for which it is selected.
This is in fact what is expressed by the first
rationality principle above. These semantics of the
effect of an action, within the scope of the rational
behavior model, permit us to ignore the problem of non-
15 predictability of the effect.

By way of example here is a simplified model (for
which it is the expression of the pre-conditions) of
the communicative act to inform on the truth of a
proposition:

20 *<i, Inform(j, ϕ)>*
 Precondition: $K(i, \phi) \wedge \neg K(i, K(j, \phi))$
 Effect: $K(j, \phi)$

25 This model is directly axiomatized within the
logical theory through the above principles of
rationality, and the following scheme (thus actions are
not pairs manipulated by a scheduling procedure like
data structures are, such as is done in the
framework of the classic guided plan approach, but have
30 a logical semantism within the theory itself):

$$K(h, Feasible(<i, Inform(j, \phi)>) \Leftrightarrow \neg K(i, \phi) \wedge K(i, K(j, \phi)))$$

Let us note that the two above principles by themselves specify (without any extra-logical artifice) a scheduling algorithm, which deductively produces action plans by inference of causal chains of intent.

10 **FORMALIZATION OF SOME PRINCIPLES OF COOPERATIVE BEHAVIOR**

You can refer to (Sadek 91a, 94a) for a detailed proposition of a model of the cooperative behavior within a formal theory of rational interaction.

15 **THE ADOPTION OF INTENTION OR THE MINIMAL PRINCIPLE OF COOPERATION**

A priori, nothing on the strictly rational plan constrains an agent to be (either largely or slightly) cooperative and, in particular, to react to prompts from others (such as for example, to respond to questions that are put to it). This limiting condition, which we call the *minimum principle of cooperation*, is a particular case of the following property of adoption of intent: If an agent *i* thinks that an agent *j* has the intention to realize a property *p*, and said agent itself does not have the opposite intent, then *i* is going to adopt the intention that *j* may know (one day) that *p* is realized. This property translates formally into the validity of the following formula scheme:

30

$$K(i, I(j, p)) \wedge \neg I(i, \neg p) \Rightarrow I(i, K(j, p))$$

Together, the two previous properties guarantee that an agent is going to act sincerely, and therefore to cooperate. Otherwise, it is important to underline that they express a lot more than a principle of minimal cooperation. In fact, they expresses a principle of "quite brief" cooperation. They translate the fact that from the moment an agent learns of the objectives of another agent, then it will help the other to attain them, insofar as they do not contradict its own objectives.

THE RELEVANCE

Most of the remarkable types of cooperative responses are manifest by the communication of supplemental of information with regard to what has been explicitly requested. However, the amount of additional information depends heavily on the presumed interest of the party requesting this information, and in particular, of its acknowledged intentions. The notion of interest is very contextual and remains rather tricky to establish in the general case. On the contrary, it is information, obviously, that are not pertinent to the interlocutor, those for example, (presuppositions) already known about him. Stated otherwise, the avoidance of redundancy is a component of cooperative behavior, which can be expressed as follows, in terms of the elementary property (that, actually, is not primitive but derives directly from

the same definition of the concept of intention): if an
i agent has the intent to make known to an agent j a
proposition p then i must think that j does not already
know it. Formally it translates from the validity of
5 the following scheme:

$$I(i, K(j,p)) \Rightarrow K(I, \neg K(j,p))$$

09869205-091701

THE ADJUSTMENT OF BELIEFS

A corrective response is generated with the intention to correct the interlocutor's belief, judged as erroneous. This belief generally constitutes a presupposition inferred (by implication (Grice 75)) from the recognized communicative act. The intention in question is generated at an agent's location every time that its belief about a proposition regarding which it does not believe its interlocutor competent, is found to be in contradiction with that of its interlocutor. It translates formally in the validity of the following scheme:

$$K(i, (p \wedge K(j, \neg p)) \Rightarrow I(i, K(j, p))$$

REACTION TO PROMPTS

In a communicating system an agent cannot resolve the non recognition by a phenomenon that it has observed.

To give an account of this feature, the following dual property is formulated: the first choice of this property stipulates that after a phenomenon that an agent perceives and with which, either it cannot associate an intelligible event, or any event that it can associate is unacceptable in view of its beliefs, the agent is going to adopt the intention to learn what has taken place, typically by generating a request for repetition. The second choice of this property, which is less general than the first, only concern the case where the agent cannot, according to its mental state, accept any feasible event whatsoever from what he